

MANAGEMENT OF VARIABLE ACCURACY FOR APPLICATION OPTIMIZATION

PROBLEM STATEMENT The immense computing needs of modern science force developers to spend significant time optimizing their applications to achieve high performance. This typically involves explicit, hardware-centric management of application parallelism, memory layout and access order. “Algorithmic optimization” is an alternative where algorithmic details are adjusted to ensure that the needed scientific results are computed with no wasted work. Although both approaches achieve excellent results, the efficiency due to algorithmic optimization is portable across systems and is managed at a high level where developers have the most insight and control. Unfortunately, unlike the significant investments made in support tools for hardware optimization, work on algorithmic optimization is almost completely manual and unsupported by tools.

STATE OF THE ART Algorithmic optimization has seen significant research work over the past several decades. For example, adaptive mesh refinement (AMR) varies the spatial resolution of different regions based on the required accuracy. The use of AMR enabled the CTH application to improve performance by a factor of 2 (1). Similarly, the use of single rather than double precision enables significantly faster communication (single takes half the storage and bandwidth) and computation (single-precision arithmetic is 2-6x faster on GPUs and 2x faster with many CPU vector vector instructions) (2) (3). Recently we showed (Figure 1) that in the ParaDiS (4) crack dislocation simulation a careful reduction in the frequency at which dislocation nodes are analyzed can improve performance by 20% while maintaining application accuracy. Further, we have demonstrated (5) a resilient sparse matrix-vector multiplication algorithm that can be reconfigured using a decision tree model for any given input matrix to provide high resilience and performance. Finally, our collaborators at LLNL have sped up the Cardiod (6) simulation by 18% by re-implementing the logarithm function to give up a controlled amount of precision and have shown (7) that a careful tuning of the parameters of the Algebraic Multi-Grid simulation in the hypre solver library enables significant performance improvements for each given input problem.

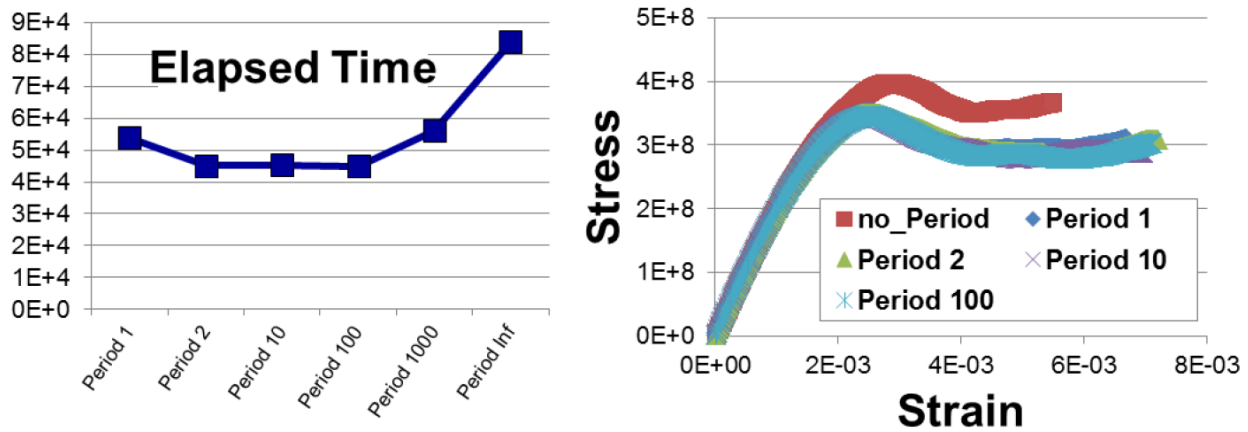


Figure 1: Varying the period of dislocation node detection improves performance without sacrificing accuracy.

APPROACH The promise of algorithmic optimization is constrained by the difficulty of reasoning about the effects of algorithm configurations on overall application accuracy. This makes it imperative to develop tools that enable developers to make this technique a routine part of their performance optimization efforts. Such a tool would enable application developers to identify major application modules and quantify the error in their outputs. This would include simple measurements of the difference between data structures in a reference application run and a run with an alternate configuration (it is assumed that there exist reference executions that configure application modules to be less efficient but more accurate), as well as more elaborate and optional online error estimators. Such information must be visualized to show the impact of various module configurations on application accuracy as well as the propagation of errors to the rest of the

application. For example, Figure 2 shows a sample view of how changes in Module A's convergence threshold affect the accuracy of subsequent application states. Such visualizations will empower developers to identify opportunities to improve application efficiency by managing module accuracy.

In addition to identifying major algorithmic optimization opportunities it is necessary to develop techniques that simplify the design and deployment of static and adaptive configuration strategies. The main challenge of this task is (i) predicting the error that will result from a given configuration decision and (ii) subsequently validating that this error is indeed within acceptable bounds. To this end two technologies are needed, as shown in Figure 3. The first is statistical modeling tools to capture error propagation based on experimental observations of representative application runs. These models would extend prior work by the Uncertainty Quantification (UQ) community and require new work on sampling strategies to leverage the fact that error propagation through individual application code modules or processors can be observed independently from others. This can enable more comprehensive error propagation studies based on fewer full application runs. Since these the predictions of the error propagation models will have some errors, it will also be necessary to develop novel error estimators to quantify the application's error during production application runs.

EXAMPLE For a concrete example of how the proposed methodology can be applied in a real application, consider the ddcMD molecular dynamics simulation. This application is divided into two modules, a force calculation and an integrator. Force calculations are divided into short- and long-distance interactions and produce a vector of forces on all particles. The integrator applies these forces to particle positions and velocities, incorporating the Vorticity-Verlains model for soft collisions, and more specific strategies for modeling hard collisions. It outputs updated particle data. Each ddcMD module can be configured to trade off efficiency for accuracy. First, all force modules have multiple numerical representations. The short-range interactions can use various cutoffs for the Taylor expansions of the pair-wise interaction routines as well as square root approximations. The long-range interactions use different mesh granularities, widths of the charge distribution stencil as well as the precisions (single vs double) used to communicate FFT data. The accuracy of the intermediate results (forces, positions and velocities) can be measured by comparing their values and the derivatives of these values to those in a reference run known to be sufficiently accurate. The overall error may be measured in various ways, such as the average across all particles of the root mean square or maximum error, or the cumulative distribution of these errors. Finally, the effects of various configuration choices can be observed at the next time step or tracked over longer time periods.

SUMMARY The proposed methodology will support application developers' efforts to optimize the algorithms within their simulations to achieve high efficiency without sacrificing accuracy. By quantifying and visualizing the effects of module configurations on the accuracy of their results and models it will support the design and deployment of static and dynamic configuration strategies. **The result will be a new generation of highly tuned and adaptable applications that efficiently utilize system resources.**

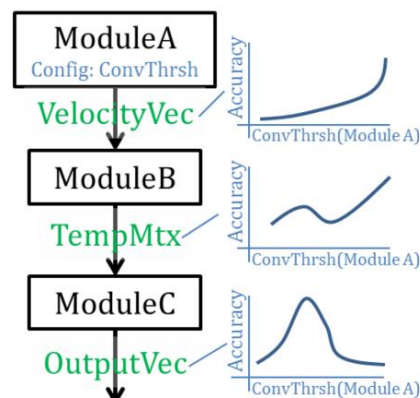


Figure 2: Accuracy visualization

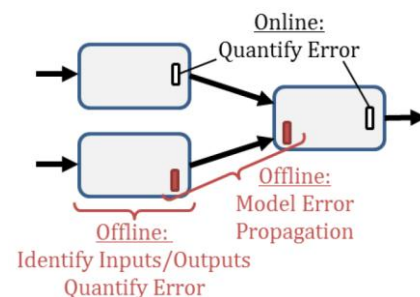


Figure 3: Measuring error propagation

BIBLIOGRAPHY

1. *Adaptive Mesh Refinement in CTH*. **Crawford, David**. 1999. US Army Symposium on Solid Mechanics.
2. *Implementation of a Mixed-Precision in Solving Systems of Linear Equations on the Cell Processor*. **Dongarra, Jakub Kurzak and Jack**. 10, 2007, Journal of Concurrency and Computation: Practice & Experience, Vol. 19, pp. 1371-1385.
3. *Accelerating Scientific Computations with Mixed Precision Algorithms*. **Marc Baboulin, Alfredo Buttarib, Jack Dongarra, Jakub Kurzak, Julie Langouc, Julien Langou, Piotr Luszczek, Stanimire Tomov**. 2008, Computer Physics Communications.
4. *Enabling Strain Hardening Simulations with Dislocation Dynamics*. **A Arsenlis, W Cai, M Tang, M Rhee, T Oppelstrup, G Hommes, G Pierce, V V Bulatov**. 2007, Modeling and Simulation in Materials Science and Engineering, Vol. 15, pp. 553-595.
5. *Algorithmic Approaches to Low Overhead Fault*. **Joseph Sloan, Rakesh Kumar and Greg Bronevetsky**. 2012. International Conference on Dependable Systems and Networks (DSN).
6. **Heller, Arnie**. Venturing into the Heart of High Performance Computing Simulations. *LLNL Science and Technology Review*. 2012.
7. *On Long Range Interpolation Operators for Aggressive Coarsening*. **Yang, Ulrike Meier**. 2-3, 2010, Numerical Linear Algebra with Applications, Vol. 17.